Search More, Think Less: Rethinking Long-Horizon Agentic Search for Efficiency and Generalization

Chengjun Yu[1,†] Shu Xu[2,†] Jiaqi Wu[3,†] Qianben Chen[4] Tianrui Qin[5] Kang Zhu[4] Qiexiang Wang[4] Jiayu Zhang[6] Xinpeng Liu[7] Xin Gui[8] Jingyi Cao[6] Yi Yao[5] Piaohong Wang[9] Dingfeng Shi[5] He Zhu[5] Tiannan Wang[4] Yuqing Wang[5] Maojia Song[10] Tianyu Zheng[11] Jian Yang[12] Jiaheng Liu[13] Minghao Liu[14] Yuchen Eleanor Jiang[5] Wangchunshu Zhou[4,*]

[1]Southwest University; [2]The Hong Kong University of Science and Technology; [3]Tsinghua University; [4]ByteDance; [5]OPPO; [6]The Chinese University of Hong Kong; [7]Peking University; [8]Beijing University of Posts and Telecommunications; [9]City University of Hong Kong; [10]Singapore University of Technology and Design; [11]IQuest Research; [12]Beihang University; [13]Nanjing University; [14]The University of Tokyo

Abstract

Recent deep research agents primarily improve performance by scaling reasoning depth, but this leads to high inference cost and latency in search-intensive scenarios. Moreover, generalization across heterogeneous research settings remains challenging. In this work, we propose $Search\ More, Think\ Less$ (SMTL), a framework for long-horizon agentic search that targets both efficiency and generalization. SMTL replaces sequential reasoning with parallel evidence acquisition, enabling efficient context management under constrained context budgets. To support generalization across task types, we further introduce a unified data synthesis pipeline that constructs search tasks spanning both deterministic question answering and open-ended research scenarios with task appropriate evaluation metrics. We train an end-to-end agent using supervised fine-tuning and reinforcement learning, achieving strong and often state of the art performance across benchmarks including BrowseComp (48.6%), GAIA (75.7%), Xbench (82.0%), and DeepResearch Bench (45.9%). Compared to Mirothinker-v1.0, SMTL with maximum 100 interaction steps reduces the average number of reasoning steps on BrowseComp by 70.7%, while improving accuracy. Our code is available at `https://github.com/OPPO-PersonalAI/SMTL`

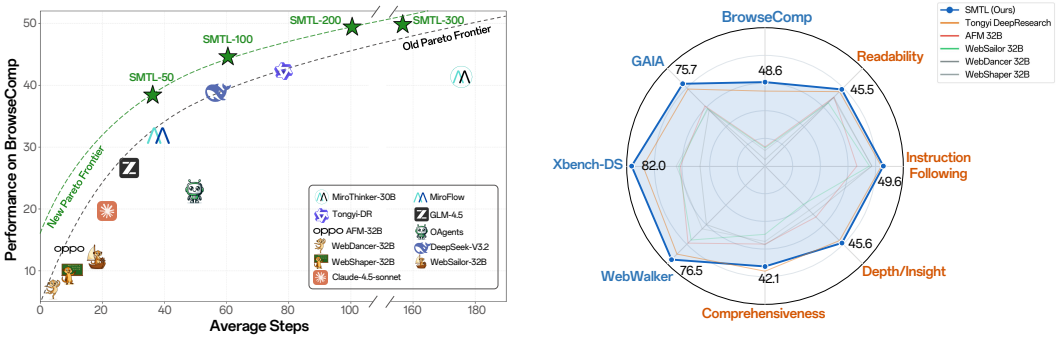
1 Introduction

Recent advances in deep research agents demonstrate that scaling reasoning depth and tool interactions can substantially improve task performance Jin et al. [18], Li et al. [26, 27], Wu et al. [59], Sun et al. [49], Zhang et al. [67], Zheng et al. [68], Zhou et al. [69, 70], Zhu et al. [71, 72], Qiu et al. [44], Roucher et al. [45], Tang et al. [51]. However, longer trajectories also lead to higher inference latency, and balancing long-horizon performance with computational efficiency remains an open problem MiroMind Team [35], Tongyi DeepResearch Team [56], Chen et al. [4], Tang et al. [50]. To train efficient long-horizon agents, a key question is: How can we design tasks that require efficient long-horizon reasoning, and construct suitable training trajectories to teach such behavior?

Furthermore, generalization across diverse evaluation settings remains challenging. Existing agentic search tasks span deterministic QA with clear ground-truth answers (e.g., BrowseComp Wei et al. [57], GAIA Mialon et al. [32], WebWalker Wu et al. [60], ACADREASON Gui et al. [14]) and open-ended research problems evaluated on coverage and synthesis quality (e.g., DeepResearch

[†]Work done during internship at OPPO Agent Team;
[*]Correspondence to: Wangchunshu Zhou <wcszhou@outlook.com>.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Efficiency on BrowseComp: score vs. interaction steps.

(b) Generalization across multiple benchmarks.

Figure 1: **Overview of SMTL Performance.** (a) Efficiency on BrowseComp. All methods are evaluated with their default settings. (b) Generalization across benchmarks. Comprehensiveness, depth, instruction following, and readability are measured following DeepResearch Bench. We report SMTL-300 for Deep Search benchmarks and SMTL-100 for Deep Research benchmarks.

Bench Du et al. [8], DeepResearchGym Coelho et al. [6]). These two settings have very different optimization objectives, making it difficult to train a single agent that performs well across both.

In this work, we revisit long-horizon agentic search from the perspectives of efficiency and generalization. We argue that the main scalability bottleneck of existing deep research agents lies in their reliance on linear, sequential reasoning in search tasks. To address this, we propose an agentic framework that replaces sequential reasoning with parallel task decomposition and concurrent tool execution, together with structured context management for efficient long-horizon inference under constrained context budgets. At the data level, we introduce an automated pipeline that synthesizes representative multi-type search tasks, reducing redundant samples and improving generalization. Trained end-to-end with supervised fine-tuning and reinforcement learning, our approach achieves state-of-the-art performance on BrowseComp (48.6), while substantially improving efficiency by reducing the number of reasoning steps by 78% and inference latency by up to 2.6×.

Our contributions are summarized as follows:

- **End-to-end learned parallel search policy.** We propose a unified agentic framework that internalizes parallel decomposition, verification, replanning, and context management within a single policy. Unlike distillation-based approaches, our method further optimizes these decisions through on-policy reinforcement learning, achieving an improved efficiency and accuracy trade-off.
- **Generalized data construction.** We introduce an automated data pipeline that constructs representative multi-type search tasks across both deterministic and open-ended settings, supporting generalization in long-horizon agentic search.
- **State-of-the-art performance.** Our method achieves state-of-the-art results on multiple deep search and deep research benchmarks, while substantially improving efficiency in terms of reasoning steps and inference latency.

2 Related Work

Agent Frameworks and Systems. Agentic workflows augment LLMs with planning, tool use, and iterative interaction for search-intensive tasks. Commercial deep research systems combine proprietary backbones with multi-step exploration and plan refinement OpenAI [38], Google [12], Anthropic [3], Perplexity Team [40]. Structured frameworks such as WebWeaver Li et al. [28] and OAgents Zhu et al. [71] adopt planner–executor workflows, while open-source templates provide reusable orchestration pipelines LangChain [21], SkyworkAI [48], Together AI [55], with MiroFlow MiroMind AI Team [34] consolidating these into a benchmark-oriented framework. Multi-agent approaches further introduce specialized roles and coordination Li et al. [22], Hong et al. [16], Qian et al. [42], Fourney et al. [10], Roucher et al. [45], Chen et al. [5], Hu et al. [17]. Despite this diversity, most workflows scale by deepening sequential reasoning, leading to limited information efficiency.

Synthetic Data Pipelines. Collecting multi-step tool-interactive trajectories at scale remains costly. Graph-based generation, as in WebSailor Li et al. [24], samples complex QA pairs from knowledge subgraphs; easy-to-hard expansion progressively extends seed questions into longer-horizon problems Tao et al. [52], Gao et al. [11], Liu et al. [31]; and TaskCraft Shi et al. [47] expands tasks along depth and width with incremental validation. However, existing pipelines primarily emphasize task difficulty rather than shaping information-efficient search behaviors, and provide limited support for open-ended research tasks requiring flexible information aggregation.

3 Parallel Agentic Workflow

Recent tool-augmented agents enable language models to interact with external tools for retrieval and reasoning Yao et al. [63], Xue et al. [62], Li et al. [27]. Building on this line of work, we design an efficient parallel agentic workflow inspired by Flash-Searcher Qin et al. [43], which explicitly supports concurrent execution and structured coordination across subtasks. As illustrated in Figure 2, the agent initializes a task plan, executes multiple subtasks in parallel, and periodically synchronizes intermediate results through plan refinement before producing the final answer.

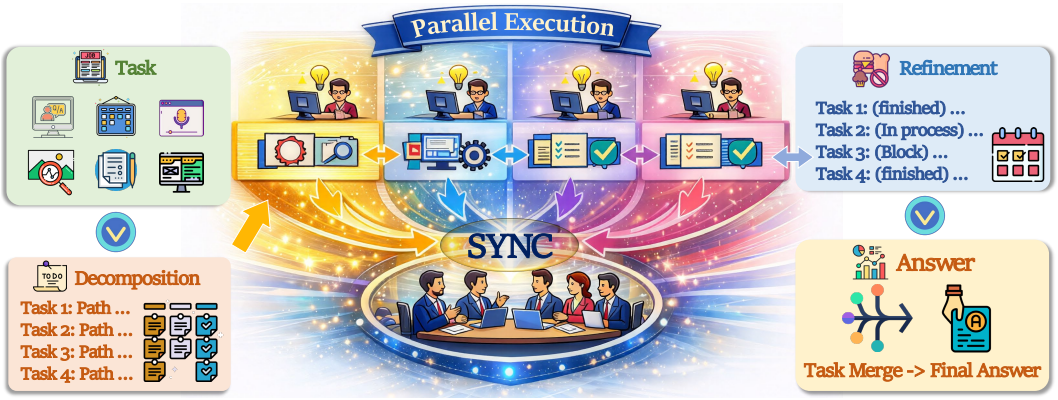


Figure 2: Overview of our parallel agentic workflow design.

Initial Plan Construction. Given a composite search task, the agent constructs an initial task plan G^0_{plan} by decomposing the problem into a set of interrelated yet partially independent subtasks, each corresponding to a concrete information-seeking or verification objective. The plan exposes parallelizable execution paths early, enabling concurrent evidence acquisition and higher information density.

Parallel Execution and Tool Coordination. At each timestep, the system selects ready-to-execute subtasks from the pending set $\mathcal{P}_t$ based on their readiness, while completed subtasks are tracked separately. These pending subtasks are processed concurrently, leveraging available tools or agent actions to gather information and execute reasoning tasks. The system aggregates observations from each parallel execution into a unified reasoning state:

$$s_{t+1} = \mathcal{F}(s_t, \{a_t^{(k)}\}_{k=1}^m, \{o_t^{(k)}\}_{k=1}^m), \tag{1}$$

where $a_t^{(k)}$ and $o_t^{(k)}$ represent the actions and observations of the k-th parallel execution, and s_t denotes the aggregated reasoning state at timestep t.

In practice, parallel execution is realized via a limited set of reusable external tools (Appendix A), including web search and page crawling, which are repeatedly invoked across pending subtasks to facilitate concurrent information acquisition and verification.

Dynamic Plan Refinement. To ensure the plan adapts to the ongoing execution, the task plan is periodically updated. Completed subtasks are removed, unresolved dependencies are rechecked, and new subtasks may be introduced. The task plan is refined based on the current execution state:

$$G^{t+\Delta}_{\text{plan}} = \mathcal{R}(G^t_{\text{plan}}, \mathcal{C}_t, \mathcal{P}_t, s_t),$$

where $\mathcal{C}_t$ represents the completed subtasks. This dynamic refinement ensures that the task adapts to progress and maintains efficiency. The complete algorithmic outline is provided in Appendix B.

4 Data Construction

Existing agent data construction pipelines limit both generalization and efficiency. Many rely on static knowledge sources and focus on deterministic, entity-centric Deep Search tasks, providing weak coverage of open-ended Deep Research scenarios. Moreover, task difficulty is often scaled by increasing reasoning hops rather than improving information density, leading to redundant evidence and inefficient interaction traces.

To address these issues, we propose a **high-diversity, high-density** data synthesis pipeline. As illustrated in Fig. 3, our framework integrates raw corpus collection, graph construction, subgraph extraction, and QA generation with verification, enabling dense and correlated evidence acquisition for both deterministic and open-ended research tasks.

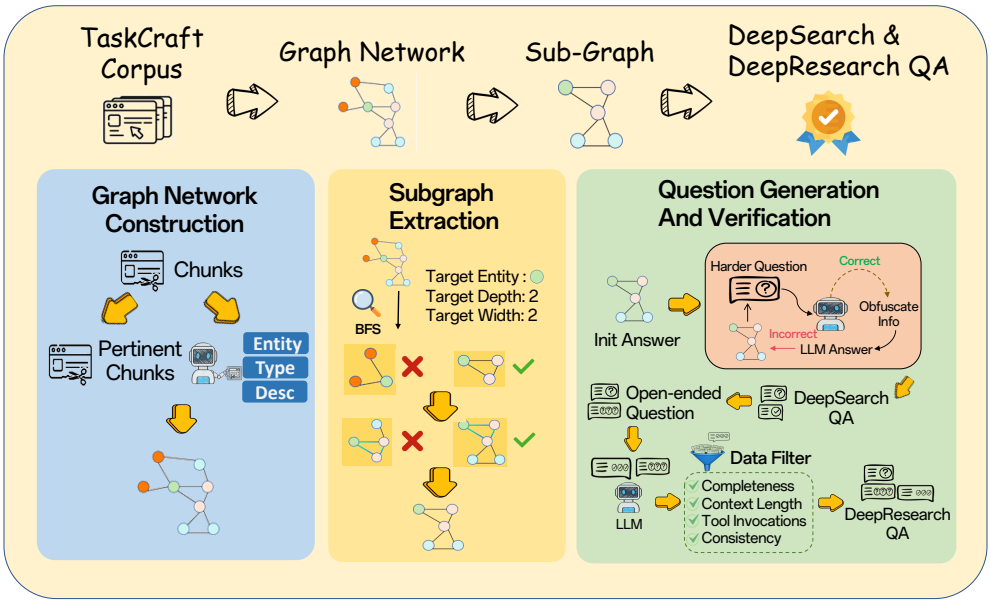


Figure 3: Overview of the data construction pipeline.

4.1 Deep Search Data

Raw Corpus Collection. To support cross-domain generalization and multi-hop reasoning, we construct the raw corpus by leveraging trajectories from TaskCraft corpus Shi et al. [47]. These trajectories contain rich collections of real-world URLs spanning diverse domains, including art, sports, history, government, economics, politics, music, geography, movies, computer science, physics, and chemistry.

Crucially, URLs within each trajectory are not independent: they are connected through explicit information-seeking paths, where later queries and sources build upon evidence gathered from earlier ones. This structure naturally induces multi-hop relationships across documents, making the collected corpus well suited for graph-based task construction. We extract and normalize the content from these URLs using document parsing tool Jina Jina [19], yielding a large-scale, high-diversity raw corpus that preserves both domain coverage and inter-document relational structure.

Graph Network Construction. Based on the initial corpus, we develop an efficient pipeline for generating sophisticated graph networks. First, we leverage LightRAG Guo et al. [15] to instantiate a knowledge-driven graph network. This process involves partitioning the curated text crops into multiple chunks, from which an LLM extracts entities and their respective attributes. By integrating

an embedding-plus-reranker retrieval mechanism, we recall pertinent chunks, enabling the LLM to synthesize detailed node descriptions and delineate complex inter-node relationships, ultimately culminating in a highly intricate graph network.

Subgraph Extraction. Given the constructed knowledge graph, we extract task-specific subgraphs using a controlled random-walk strategy. For each task, we sample a target entity as the ground-truth answer and perform a breadth-first search (BFS) up to N hops to collect its surrounding neighborhood. The resulting subgraph defines the supporting evidence structure required to infer the answer, where multi-hop nodes serve as question conditions with varying degrees of indirection. By adjusting the hop depth and branching factor, we flexibly control task difficulty while preserving semantic coherence and factual correctness. This construction yields compact, information-dense task skeletons that require integrating multiple related evidence sources rather than relying on single-hop retrieval.

To ensure high-quality and non-degenerate task structures, we adopt the following design principles:

- **Rich topological structures.** We prioritize subgraphs in which two $(N+1)$-hop nodes are interrelated while sharing a common N-hop parent, inducing cyclic dependencies that require cross-validation of multiple relationships.
- **Controlled walk width and depth.** We explicitly bound the depth and branching factor to keep task difficulty scalable and avoid trivial shortcuts or excessively long reasoning chains.

Hierarchical Question Construction and Verification. Given a task-specific subgraph with a fixed target answer, we construct questions through a hierarchical synthesis process. Starting from the outermost N-hop frontier, we iteratively aggregate information from $(i+1)$-hop nodes to form sub-questions about i-hop entities. Each aggregation step yields a valid intermediate question, and progressively merging all layers produces a final question about the target entity that requires the maximum hop depth and reasoning difficulty.

When multiple $(i+1)$-hop nodes exhibit semantic relationships, we explicitly encode these interdependencies as verifiable conditions, requiring agents to cross-validate parallel evidence paths rather than rely on linear reasoning. To prevent information leakage, we apply an LLM-based verification step after each synthesis iteration; if the answer can be inferred prematurely, the question is restructured or relevant information is obfuscated. This process repeats until the desired difficulty is achieved or a maximum of five iterations is reached.

The complete designs of the prompts for entity information extraction, entity evaluation, entity question generation and obfuscation, and verification are described in Section H.1.

4.2 Deep Research Data

Research Question Construction. Deep research tasks are synthesized entirely within our unified data construction pipeline, without relying on externally curated queries. Given a task-specific subgraph with a fixed target entity and its multi-hop supporting structure, we formulate open-ended research questions that require integrating evidence across the entire subgraph. These questions are designed to elicit report-style answers involving explanation, comparison, and synthesis across multiple sources, rather than single factual outputs. This construction explicitly encourages long-horizon planning, information aggregation, and cross-source reasoning.

Trajectory Generation and Quality Filtering. For each open-ended research question, we generate multiple candidate trajectories using our parallel agentic workflow and apply a two-stage filtering process to ensure high-quality supervision. First, trajectories are subjected to rule-based hard rejection to remove structurally invalid or overly shallow solutions, enforcing basic requirements on completeness, context validity, reasoning depth, tool usage, and format consistency. Second, trajectories that pass these checks are then evaluated by an LLM-as-a-Judge, which assesses higher-level semantic quality, including comprehensiveness, insight/depth, instruction following, and readability. Only trajectories that satisfy both structural constraints and semantic criteria are retained as reference outputs, yielding scalable and reliable training data for open-ended deep research tasks.

The complete designs of the prompts for open-ended question construction and quality filtering are illustrated in Section H.2.

Table 1: Main results on Deep Search and Deep Research benchmarks. For Deep Search benchmarks, BC: BrowseComp; Xbench-DS: Xbench-DeepSearch; WW: WebWalker-QA; DR-Gym: DeepResearch Gym; DR-Bench: DeepResearch Bench. For Deep Research Bench, Comp., Depth, Inst., and Read. denote Comprehensiveness, Insight/Depth, Instruction Following, and Readability, respectively. We report pass@1 for our models. Results marked with * are obtained by our own evaluation, while results marked with $^+$ are taken from prior work Yao et al. [64].

Model	Deep Search						Deep Research Bench RACE				
	BC	GAIA	Xbench-DS	WW	FRAMES	SEAL-0	Overall.	Comp.	Depth	Inst.	Read.
Foundation Models with Tools											
GLM-4.5	26.4	66.0	70.0	65.6	78.9	36.0	–	–	–	–	–
Minimax-M2	44.0	75.7	72.0	64.5	–	–	46.1	45.2	44.6	49.0	44.7
DeepSeek-V3.2	40.1	63.5	71.0	–	80.2	38.5	–	–	–	–	–
Kimi-K2-0905	14.1	60.2	61.0	63.0	58.1	25.2	44.5	42.8	39.7	50.8	46.0
Claude-4.5-Sonnet	19.6	71.2	66.0	–	85.0	53.4	39.9	–	–	–	–
GPT-5	54.9	59.4	–	73.0	90.0	51.4	46.8	45.4	44.5	50.3	47.5
Gemini-2.5-Pro	-	60.2	56.0	–	–	19.8	35.1	34.1	29.8	41.7	37.2
Deep Research System											
OpenAI DeepResearch	51.5	67.4	26.6	–	–	–	47.0$^+$	46.9$^+$	45.3$^+$	49.3$^+$	47.1$^+$
Gemini DeepResearch	59.2	–	50.0	–	–	–	48.9$^+$	48.5$^+$	48.5$^+$	49.2$^+$	49.4$^+$
Perplexity Deep Research	22.0	–	–	67.0	83.0	38.7	42.3$^+$	40.7$^+$	39.4$^+$	46.4$^+$	44.3$^+$
Kimi-Researcher	26.9	–	69.0	–	78.8	36.0	44.6	45.0	42.0	47.1	45.6
OAgents (Claude-3-7)	22.2	66.7	54.5	53.0	–	–	50.8$^+$	50.4$^+$	51.2$^+$	50.3$^+$	49.4$^+$
MiroFlow (GPT-5)	33.2	82.4	72.0	52.6	–	–	44.9	44.6	49.3	45.6	46.1
Open-source Agentic Model											
WebSailor-32B	10.5	53.2	53.3	60.5	69.8	16.2	32.4	28.6	22.7	43.7	37.5
WebDancer-QwQ	3.8	51.5	40.0	47.9	–	20.7	35.9*	33.0*	28.5*	44.8*	40.8*
WebShaper-32B	9.0	52.4	54.6	51.4	–	–	34.9	31.6	26.2	44.8	40.4
DeepMiner-32B-RL	33.5	58.7	62.0	–	–	–	–	–	–	–	–
AFM-32B-RL	11.1	55.3	52.0	63.0	–	–	35.8*	32.7*	30.2*	38.5*	40.9*
NestBrowse-30B-A3B	31.6	**75.7**	75.0	–	–	–	–	–	–	–	–
Tongyi-DeepResearch-30B	43.4	70.9	75.0	72.2	**90.6**	–	45.7$^+$	**44.7**$^+$	44.2$^+$	48.8$^+$	44.2$^+$
MiroThinker-v1.0-30B	41.2	73.5	70.6	61.0	85.4	46.8	–	–	–	–	–
SMTL-100	43.6	74.8	80.0	74.9	84.3	50.5	**45.9**	42.1	**45.6**	**49.6**	**45.5**
SMTL-300	**48.6**	**75.7**	**82.0**	**76.5**	85.1	**51.4**	-	-	-	-	

5 Training Recipe

5.1 Post-training Supervised Fine-tuning.

We perform supervised fine-tuning (SFT) to initialize the agent with stable and efficient search behaviors before reinforcement learning.

Task Composition. The SFT dataset consists of two task categories: *Deep Search* and *Deep Research*, which differ in supervision form while sharing the same underlying subgraph-based construction.

- **Deep Search.** Tasks are instantiated from task-specific subgraphs with hop depth ranging from 2 to 5. For each subgraph, all hierarchical question variants constructed during iterative aggregation are retained, yielding multiple questions that share the same target entity as the ground-truth answer. To prevent over-representation of frequent answers, we apply an answer frequency threshold and discard tasks whose target entities appear excessively often.
- **Deep Research.** For each subgraph, we construct an open-ended research question centered on the target entity and its multi-hop supporting structure. Questions are formulated to encourage broad exploration and synthesis over the entire subgraph, rather than single-answer retrieval, ensuring sufficient topical richness and variability.

Trajectory Construction and Curation. Training trajectories are generated using the agentic workflow described in Section 3. Deep Search trajectories are distilled from DeepSeek-V3.2 Liu et al. [29], while Deep Research trajectories are distilled from GPT-5 OpenAI [37]. We apply quality curation including context length capping (64K tokens), minimum parallel tool-call density ($\geq$3 calls/step), and shortest-correct-trajectory selection. Details are provided in Appendix D.

5.2 Reinforcement Learning.

We adopt a modified REINFORCE Leave-One-Out (RLOO) algorithm Ahmadian et al. [1] with token-level loss Yu et al. [65] and sequence-level importance sampling for rollout correction Liu et al. [30]. We additionally filter negative trajectories caused by environmental errors or excessive length to stabilize training. The reward is outcome-based: an LLM judge assigns 1 for correct final answers and 0 otherwise; format-violating tool calls trigger immediate termination with reward 0. Full details of the RL modifications are in Appendix D.

6 Experiments

6.1 Experimental Setup

We compare SMTL against three categories of systems: foundation models with tools, deep research systems, and open-source agentic models. Evaluation covers both deep search benchmarks (BrowseComp, GAIA, XBench-DeepSearch, WebWalkerQA, FRAMES, SEAL-0) and deep research benchmarks (Deep Research Bench RACE). Full details on baselines, benchmarks, and evaluation metrics are provided in Appendix C.

Implementation Details. We conduct experiments using Qwen3-30B-A3B-Instruct-2507 as the backbone. Models are trained with SFT followed by RL under the agentic workflow described in earlier sections. During inference, we use vLLM with a 128K context window and a default maximum of 100 interaction steps with plan refinement every $N{=}5$ steps. Full training hyperparameters are provided in Appendix D.

Context Management during Inference Stage. Long-horizon tasks often exceed the 128K context window, especially under SMTL's parallel execution which produces more observations per step. To address this, SMTL couples periodic plan refinement (every $N{=}5$ steps) with an overflow-triggered compression: when context reaches the budget limit, the agent performs a forced plan refinement, then drops all pre-plan history and continues from the refreshed plan. This plan-centric reset preserves execution state while enabling longer effective trajectories. We evaluate SMTL under maximum step limits of 100 and 300 (SMTL-100, SMTL-300); for Deep Research Bench, we report the 100-step setting. Further details are in Appendix F.

6.2 Main Results

SMTL exhibits consistent Pareto dominance across heterogeneous benchmarks. As shown in Table 1, SMTL-30B demonstrates strong performance across Deep Search benchmarks under different interaction budgets. With a moderate budget (SMTL-100), the model already achieves state-of-the-art performance among 30B-scale open-source agentic models on BrowseComp (43.6%), slightly surpassing Tongyi-DeepResearch-30B (43.4%) and clearly outperforming MiroThinker-v1.0-30B (41.2%). It also reaches 78.0% on XBench-DeepSearch and 74.9% on WebWalker-QA. When the budget is increased to 300 steps, performance further improves, most notably on the long-horizon benchmark BrowseComp, where accuracy rises from 43.6% to **48.6%** (+5.0), substantially widening the gap over both Tongyi and MiroThinker. In contrast, gains on shorter-horizon tasks such as GAIA (74.8% $\rightarrow$ **75.7%**) and WebWalker (74.9% $\rightarrow$ **76.5%**) are comparatively modest, indicating that additional interaction budget primarily benefits deeper multi-step evidence aggregation. Moreover, as shown in Figure 1a, across interaction budgets from 50 to 300 steps on BrowseComp, SMTL consistently lies on the Pareto frontier of accuracy versus trajectory length, while mainstream baselines fall below this curve, confirming its superior efficiency-aware scaling on long-horizon search.

Beyond deterministic deep search, SMTL further generalizes to open-ended deep research evaluation. On DeepResearch Bench RACE, SMTL-100 achieves an overall score of **45.9%**, with strong and balanced performance across Comprehensiveness (42.1%), Insight/Depth (45.6%), Instruction Following (49.6%), and Readability (45.5%). This surpasses representative open-source agentic baselines such as WebSailor-32B (32.4%), WebDancer-QwQ (35.9%), WebShaper-32B (34.9%), and AFM-32B-RL (35.8%), and slightly outperforms Tongyi-DeepResearch-30B (45.7%) and Kimi-Researcher (44.6%), establishing strong competitiveness among 30B-scale systems. This demonstrates that the same parallel search framework transfers from accuracy-driven benchmarks to long-form research synthesis without task-specific modification. Furthermore, as illustrated in Figure 1b, SMTL consistently achieves leading average performance across benchmarks, confirming that its gains are not confined to a single task but reflect robust generalization across diverse objectives and evaluation criteria.

Why SMTL is more efficient? We analyze the efficiency advantage of SMTL through a qualitative case study comparing SMTL-30B with MiroThinker-v1.0-30B on a BrowseComp task. As illustrated in Figure 5, SMTL localizes the key entity within 8 assistant turns, whereas MiroThinker-v1.0 requires 16 turns. SMTL decomposes the task into multiple hypothesized subtasks explored in parallel, rapidly surfacing high-signal evidence and periodically re-planning based on intermediate observations. In contrast, MiroThinker follows a strictly sequential pattern with one tool call per round, requiring repeated query reformulation and delaying key evidence discovery. This demonstrates that SMTL's efficiency gains stem from parallel subtask exploration and staged re-planning rather than deeper per-step reasoning.

7 Analysis

7.1 Efficiency Evaluation

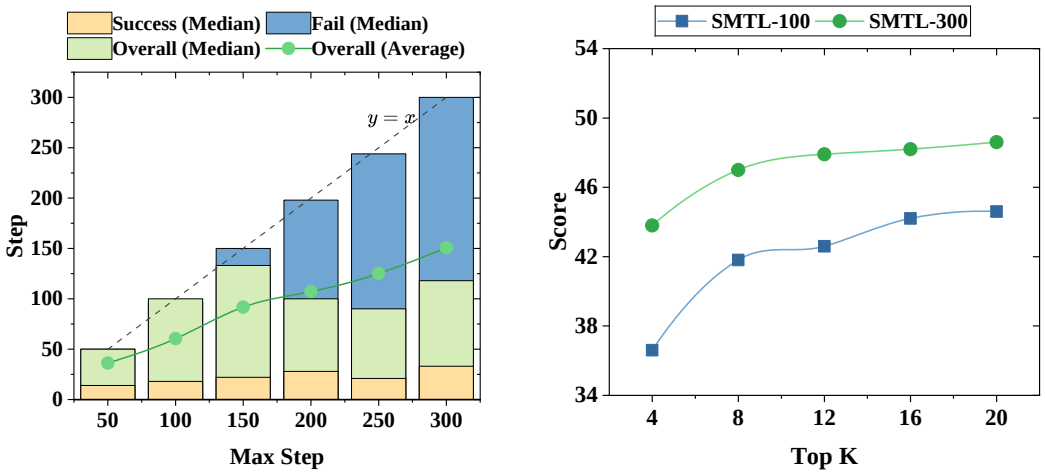
We evaluate the efficiency of SMTL from the perspective of interaction complexity, measured by the number of assistant steps required to solve a task. As shown in Table 2, SMTL achieves a more favorable balance between interaction cost and task performance compared to representative baselines on BrowseComp. SMTL-100 attains 44.6% accuracy with an average of 60.4 assistant steps, slightly outperforming Tongyi-DeepResearch-30B (43.4%) while requiring fewer steps (75.2). The contrast with MiroThinker-v1.0-30B is even more pronounced: MiroThinker requires 206.0 steps to reach 41.2% accuracy, whereas SMTL-100 achieves substantially higher accuracy with less than one-third of the interaction cost.

This efficiency is closely related to SMTL's parallel execution mechanism. Unlike sequential systems that invoke a single tool per round, SMTL performs an average of 3.5 tool calls per step, enabling concurrent evidence acquisition across subtasks. By aggregating more information within each interaction round, SMTL increases information density per step and reduces redundant query reformulation, resulting in shorter yet more effective trajectories.

Increasing the interaction budget to SMTL-300 further improves accuracy to 48.6%, with 150.7 average steps. While the trajectory length grows as expected, the additional steps translate into measurable performance gains rather than inefficient looping. Figure 1a further illustrates that across different interaction budgets, SMTL consistently lies on the Pareto frontier in the accuracy–step plane, confirming that its performance improvements are achieved with well-controlled interaction complexity.

Table 2: **Interaction efficiency on BrowseComp.** Average number of assistant steps, average tool calls per step, and task accuracy.

Model	Steps	Avg. Tool Calls	Acc.
Tongyi-DeepResearch-30B	75.2	1.0	43.4
MiroThinker-v1.0-30B	206.0	1.0	41.2
SMTL-100	**60.4**	3.5	44.6
SMTL-300	150.7	3.7	**48.6**



(a) BC score under different max steps.

(b) BC score under different web search topk results.

Figure 4: Results of context management (CM) under different observation horizons K with interaction budgets (IB) of 80 and 160 steps on the BC benchmark.

7.2 Ablations on Maximum Interaction Steps.

We conduct ablation studies to analyze how the maximum interaction budget affects trajectory behavior on BrowseComp, varying max steps from 50 to 300. As shown in Figure 4a, the median step count of successful cases remains roughly constant regardless of budget, indicating that correct trajectories converge early. In contrast, the median steps of failed cases closely follows $y{=}x$, showing that most failures exhaust the full budget rather than terminating prematurely. This suggests the model actively explores alternative paths when facing difficulty. When the budget is enlarged, SMTL gains additional opportunities to explore different evidence chains and, combined with periodic plan refinement, can correct suboptimal search directions. Thus, increasing max steps primarily benefits genuinely hard cases by alleviating search budget constraints rather than compensating for systematic reasoning errors.

7.3 Ablations on Retrieval Top-k.

We investigate how retrieval width affects performance by varying top-k (number of URLs per query) for both SMTL-100 and SMTL-300 on BrowseComp. As shown in Figure 4b, increasing top-k consistently improves accuracy. The gain is most pronounced from $k{=}4$ to $k{=}8$ (e.g., SMTL-300 improves from 43.8 to 47.0, while SMTL-100 increases from 36.6 to above 41.8), indicating that a narrow retrieval window significantly constrains evidence coverage. Beyond $k{=}8$, performance continues to improve but with diminishing returns as the most informative candidates are already captured. These results confirm that expanding retrieval breadth is an effective scaling axis for long-horizon search: under a fixed interaction budget, larger top-k packs more candidate evidence into each search action, raising information density per step rather than requiring deeper reasoning chains.

8 Conclusion

We proposed SMTL, a unified agentic framework for long-horizon search that prioritizes parallel evidence acquisition over sequential reasoning. Trained end-to-end, SMTL achieves state-of-the-art performance across diverse benchmarks while substantially reducing inference latency. Our results demonstrate that prioritizing efficient, search-centric scaling over ever-deeper reasoning provides a practical and generalizable foundation for future deep research agents, and we hope this work encourages further exploration of efficiency-oriented agentic designs.

Acknowledgments

This work is supported by the Personalized AI Lab of OPPO.

References

[1] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL https://arxiv.org/abs/2402.14740.

[2] Anthropic. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5, 2025.

[3] Anthropic. Claude takes research to new places. https://www.anthropic.com/news/research, 2025.

[4] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[5] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6, 2023.

[6] João Coelho, Jingjie Ning, Jingyuan He, Kangrui Mao, Abhijay Paladugu, Pranav Setlur, Jiahe Jin, Jamie Callan, João Magalhães, Bruno Martins, et al. Deepresearchgym: A free, transparent, and reproducible evaluation sandbox for deep research. *arXiv preprint arXiv:2505.19253*, 2025.

[7] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

[8] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025.

[9] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents, 2025. URL https://arxiv.org/abs/2506.11763.

[10] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

[11] Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

[12] Google. Gemini deep research: Your personal research assistant. https://gemini.google/overview/deep-research/, 2024.

[13] Google. Gemini deep research. https://gemini.google/us/overview/deep-research/?hl=en, 2025.

[14] Xin Gui, King Zhu, JinCheng Ren, Qianben Chen, Zekun Moore Wang, Yizhi LI, Xinpeng Liu, Xiaowan Li, Wenli Ren, Linyu Miao, Tianrui Qin, Ziqi Shu, He Zhu, Xiangru Tang, Dingfeng Shi, Jiaheng Liu, Yuchen Eleanor Jiang, Minghao Liu, Ge Zhang, and Wangchunshu Zhou. Acadreason: Exploring the limits of reasoning models with academic research problems, 2025. URL https://arxiv.org/abs/2510.11652.

[15] Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*, 2024.

[16] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations, ICLR*, 2024.

[17] Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, Zeyu Zhang, Yifeng Wang, Qianshuo Ye, Bernard Ghanem, Ping Luo, and Guohao Li. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation, 2025. URL https://arxiv.org/abs/2505.23885.

[18] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[19] Inc. Jina. Jina reader, 2025. URL https://jina.ai/reader/.

[20] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4745–4759, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.243. URL https://aclanthology.org/2025.naacl-long.243/.

[21] LangChain. Open deep research. https://github.com/langchain-ai/open_deep_research, 2025.

[22] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[23] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, Weizhou Shen, Junkai Zhang, Dingchu Zhang, Xixi Wu, Yong Jiang, Ming Yan, Pengjun Xie, Fei Huang, and Jingren Zhou. Websailor: Navigating super-human reasoning for web agent, 2025. URL https://arxiv.org/abs/2507.02592.

[24] Kuan Li, Zhongwang Zhang, Huifeng Yin, Liwen Zhang, Litu Ou, Jialong Wu, Wenbiao Yin, Baixuan Li, Zhengwei Tao, Xinyu Wang, et al. Websailor: Navigating super-human reasoning for web agent. *arXiv preprint arXiv:2507.02592*, 2025.

[25] Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, Hongxuan Lu, Tianrui Qin, Chenghao Zhu, Yi Yao, Shuying Fan, Xiaowan Li, Tiannan Wang, Pai Liu, King Zhu, He Zhu, Dingfeng Shi, Piaohong Wang, Yeyi Guan, Xiangru Tang, Minghao Liu, Yuchen Eleanor Jiang, Jian Yang, Jiaheng Liu, Ge Zhang, and Wangchunshu Zhou. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl, 2025. URL https://arxiv.org/abs/2508.13167.

[26] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. *arXiv preprint arXiv:2501.05366*, 2025.

[27] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025.

[28] Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*, 2025.

[29] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

[30] Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, September 2025. URL https://richardli.xyz/rl-collapse.

[31] Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025.

[32] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

[33] MiniMax. Minimax m2 & agent: Ingenious in simplicity. https://www.minimax.io/news/minimax-m2, 2025.

[34] MiroMind AI Team. Miroflow: A high-performance open-source research agent framework. https://github.com/MiroMindAI/MiroFlow, 2025.

[35] MiroMind Team. Mirothinker: Pushing the performance boundaries of open-source research agents via model, context, and interactive scaling. *arXiv preprint arXiv:2511.11793*, 2025.

[36] Moonshot. Kimi-researcher: End-to-end rl training for emerging agentic capabilities. https://moonshotai.github.io/Kimi-Researcher/, 2025.

[37] OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, 2025.

[38] OpenAI. Introducing deep research. https://openai.com/index/introducing-deep-research/, 2025.

[39] Perplexity. Introducing perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, 2025.

[40] Perplexity Team. Perplexity deep research. https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research, 2025.

[41] Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. Sealqa: Raising the bar for reasoning in search-augmented language models, 2025. URL https://arxiv.org/abs/2506.01062.

[42] Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

[43] Tianrui Qin, Qianben Chen, Sinuo Wang, He Xing, King Zhu, He Zhu, Dingfeng Shi, Xinxin Liu, Ge Zhang, Jiaheng Liu, et al. Flash-searcher: Fast and effective web agents via dag-based parallel execution. *arXiv preprint arXiv:2509.25301*, 2025.

[44] Jiahao Qiu, Xuan Qi, Tongcheng Zhang, Xinzhe Juan, Jiacheng Guo, Yifu Lu, Yimin Wang, Zixin Yao, Qihan Ren, Xun Jiang, et al. Alita: Generalist agent enabling scalable agentic reasoning with minimal predefinition and maximal self-evolution. *arXiv preprint arXiv:2505.20286*, 2025.

[45] Aymeric Roucher, Albert Villanova del Moral, Thomas Wolf, Leandro von Werra, and Erik Kaunismäki. 'smolagents': a smol library to build great agentic systems. https://github.com/huggingface/smolagents, 2025.

[46] Inc. Serper. Serper api, 2025. URL https://serper.dev/.

[47] Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Yang, et al. Taskcraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*, 2025.

[48] SkyworkAI. Deepresearchagent: A hierarchical multi-agent system for deep research. https://github.com/SkyworkAI/DeepResearchAgent, 2025.

[49] Shuang Sun, Huatong Song, Yuhao Wang, Ruiyang Ren, Jinhao Jiang, Junjie Zhang, Fei Bai, Jia Deng, Wayne Xin Zhao, Zheng Liu, et al. Simpledeepsearcher: Deep information seeking via web-powered reasoning trajectory synthesis. *arXiv preprint arXiv:2505.16834*, 2025.

[50] Qiaoyu Tang, Hao Xiang, Le Yu, Bowen Yu, Yaojie Lu, Xianpei Han, Le Sun, WenJuan Zhang, Pengbo Wang, Shixuan Liu, et al. Beyond turn limits: Training deep search agents with dynamic context window. *arXiv preprint arXiv:2510.08276*, 2025.

[51] Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. Agent kb: Leveraging cross-domain experience for agentic problem solving. In *ICML 2025 Workshop on Collaborative and Federated Agentic Workflows*, 2025.

[52] Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

[53] Zhengwei Tao, Jialong Wu, Wenbiao Yin, Junkai Zhang, Baixuan Li, Haiyang Shen, Kuan Li, Liwen Zhang, Xinyu Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webshaper: Agentically data synthesizing via information-seeking formalization, 2025. URL https://arxiv.org/abs/2507.15061.

[54] Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

[55] Together AI. Open deep research. https://www.together.ai/blog/open-deep-research, 2025.

[56] Tongyi DeepResearch Team. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*, 2025.

[57] Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

[58] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Webdancer: Towards autonomous information seeking agency, 2025. URL https://arxiv.org/abs/2505.22648.

[59] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Yong Jiang, Pengjun Xie, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025.

[60] Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025.

[61] Xbench-Team. Xbench-deepsearch, 2025. URL https://xbench.org/agi/aisearch.

[62] Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. https://simpletir.notion.site/report, 2025. Notion Blog.

[63] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[64] Yi Yao, He Zhu, Piaohong Wang, Jincheng Ren, Xinlong Yang, Qianben Chen, Xiaowan Li, Dingfeng Shi, Jiaxian Li, Qiexiang Wang, et al. O-researcher: An open ended deep research model via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2601.03743*, 2026.

[65] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[66] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

[67] Dingchu Zhang, Yida Zhao, Jialong Wu, Baixuan Li, Wenbiao Yin, Liwen Zhang, Yong Jiang, Yufeng Li, Kewei Tu, Pengjun Xie, et al. Evolvesearch: An iterative self-evolving search agent. *arXiv preprint arXiv:2505.22501*, 2025.

[68] Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025.

[69] Wangchunshu Zhou, Yuchen Eleanor Jiang, Long Li, Jialong Wu, Tiannan Wang, Shi Qiu, Jintian Zhang, Jing Chen, Ruipu Wu, Shuai Wang, et al. Agents: An open-source framework for autonomous language agents. *arXiv preprint arXiv:2309.07870*, 2023.

[70] Wangchunshu Zhou, Yixin Ou, Shengwei Ding, Long Li, Jialong Wu, Tiannan Wang, Jiamin Chen, Shuai Wang, Xiaohua Xu, Ningyu Zhang, et al. Symbolic learning enables self-evolving agents. *arXiv preprint arXiv:2406.18532*, 2024.

[71] He Zhu, Tianrui Qin, King Zhu, Heyuan Huang, Yeyi Guan, Jinxiang Xia, Yi Yao, Hanhao Li, Ningning Wang, Pai Liu, Tianhao Peng, Xin Gui, Xiaowan Li, Yuhui Liu, Yuchen Eleanor Jiang, Jun Wang, Changwang Zhang, Xiangru Tang, Ge Zhang, Jian Yang, Minghao Liu, Xitong Gao, Jiaheng Liu, and Wangchunshu Zhou. Oagents: An empirical study of building effective agents, 2025. URL `https://arxiv.org/abs/2506.15741`.

[72] King Zhu, Hanhao Li, Siwei Wu, Tianshun Xing, Dehua Ma, Xiangru Tang, Minghao Liu, Jian Yang, Jiaheng Liu, Yuchen Eleanor Jiang, Changwang Zhang, Chenghua Lin, Jun Wang, Ge Zhang, and Wangchunshu Zhou. Scaling test-time compute for llm agents, 2025. URL `https://arxiv.org/abs/2506.12928`.

A Tools Setup

Our agent operates with a minimal yet expressive toolset designed to support web-scale information acquisition and consolidation. Specifically, we employ two core tools: a search interface for candidate retrieval and a page-level crawler for content extraction and goal-directed summarization.

- `web_search`**.** This tool provides access to web search functionality through the Serper API, which interfaces with the Google Search engine [46]. Given a model-generated query string, the tool retrieves a ranked list of search results, with the default setting returning the top five entries. Each result consists of a page title, a short snippet, and the corresponding URL. The search results serve as high-level signals for identifying potentially relevant sources and guiding subsequent crawling decisions.
- `crawl_page`**.** This tool is responsible for fine-grained content acquisition and structured summarization. It takes as input both a target URL and an explicit *goal* describing the information need to be addressed. The URL is crawled using the Jina Reader API [19], after which the retrieved page content is summarized by the DeepSeek-V3.2 model [29]. Crucially, the goal specification provides semantic guidance for the summarization process, steering the model to extract and condense information that is directly relevant to the current subtask rather than producing a generic page summary. This goal-conditioned summarization enables more targeted evidence collection and reduces irrelevant context propagation. The prompt template used for page summarization is provided in Appendix J.2.

B Algorithmic Details

The parallel agentic workflow described in Section 3 is summarized in Algorithm 1. This method continuously refines the task plan while executing multiple subtasks in parallel, ensuring efficient task completion through dynamic updates and parallel execution.

Algorithm 1 Parallel agentic workflow.

Require: Composite task T
Construct initial task plan $G^0_{\text{plan}} \leftarrow \mathcal{D}(T)$
Initialize reasoning state s_0, pending subtasks $\mathcal{P}_0 \leftarrow G^0_{\text{plan}}$, completed subtasks $\mathcal{C}_0 \leftarrow \emptyset$
while $\mathcal{P}_t \neq \emptyset$ **do**
  Select executable subtasks $\mathcal{E}_t \subseteq \mathcal{P}_t$
  Execute subtasks in $\mathcal{E}_t$ in parallel using available tools
  Update reasoning state $s_{t+1} = \mathcal{F}(s_t, \{a_t^{(k)}\}, \{o_t^{(k)}\})$
  Mark completed subtasks and update $\mathcal{C}_{t+1}$
  Remove completed subtasks from $\mathcal{P}_{t+1}$
  if $t \bmod \Delta = 0$ **then**
    Refine task plan using current execution state
    $G^{t+1}_{\text{plan}} \leftarrow \mathcal{R}(G^t_{\text{plan}}, \mathcal{C}_{t+1}, \mathcal{P}_{t+1}, s_{t+1})$
  end if
end while
Return final reasoning state s_T

C Experimental Setup Details

Baselines. For comparison, we conducted a comprehensive evaluation of our SMTL model against three categories of systems: (1) foundation models with tools, including closed-source models such as Claude-4.5-Sonnet [2], OpenAI-GPT-5 [37], Gemini-2.5-Pro [7], and open-source models GLM-4.5 [66], Minimax-M2 [33], DeepSeek-V3.2 [29], Kimi-K2-0905 [54]; (2) deep research system, including OpenAI DeepResearch [38], Gemini DeepResearch [13], Perplexity Deep Research [39], Kimi-Researcher [36], OAgents (Claude-3-7) [71], MiroFlow (GPT-5) [34]; and (3) open-source agentic models, including WebSailor-32B [23], WebDancer-QwQ [58], WebShaper-32B [53], DeepMiner-32B-RL [50], AFM-32B-RL [25], Tongyi DeepResearch-30B [56], and MiroThinker-v1.0-30B [35].

Evaluation Benchmarks. We evaluate all models' performance on a broad set of agentic benchmarks covering both deep search and deep research scenarios. Specifically, the deep search benchmarks include BrowseComp [57], GAIA [32], XBench-DeepSearch [61], WebWalkerQA [60], FRAMES [20], and SEAL-0 [41]. For deep research evaluation, we use Deep Research Bench RACE [9], which evaluates long-form, open-ended research reports and reports both an overall score and four fine-grained criteria: Comprehensiveness, Insight, Instruction Following, and Readability.

Metrics. We use LLM-as-judge approach for evaluation. For deep search tasks, we adopt the pass@1 metric with a specific judge prompt. For Deep Research Bench RACE, which assesses report quality across four criteria: Comprehensiveness (coverage of key aspects), Insight/Depth (analytical novelty), Instruction-Following (adherence to query constraints), and Readability (clarity and coherence), we use another judge prompt. See the two judge prompts in Appendix J.3.

D Implementation and Training Details

Training Hyperparameters. During supervised fine-tuning, we train the models for 3.5 epochs with a batch size of 128, using the AdamW optimizer and a cosine decay learning rate schedule with an initial learning rate of 1.4×10^{-5}. The maximum sequence length is set to 65,536 tokens to support long-horizon trajectories. All models are trained under the same agentic workflow and data settings described in the main text. In the RL stage, the learning rate is set to 1×10^{-6} with a batch size of 32. For each question, 8 on-policy rollouts are generated, with a maximum sequence length of 128K tokens, up to 120 interaction turns, and training is performed for 60 steps. During inference, we use vLLM with a context window of 128K tokens.

SFT Trajectory Curation. To ensure high-quality supervision, we apply the following curation criteria:

- **Context length constraint.** The total trajectory length is capped at 64K tokens to reduce redundant interactions and noisy supervision.
- **Interaction efficiency constraint.** The average number of tool calls per step must be no less than 3, encouraging active information acquisition.
- **Trajectory efficiency optimization.** For tasks with multiple successful trajectories, we retain only those that are correct and shortest in interaction length.

Reinforcement Learning Modifications. We adopt a modified REINFORCE Leave-One-Out (RLOO) algorithm Ahmadian et al. [1]. Compared to GRPO, RLOO provides an unbiased advantage estimator. Our modifications are as follows:

1. Following DAPO Yu et al. [65], we employ a token-level loss function.
2. To mitigate the training–inference mismatch arising from discrepancies between the inference engine and the training framework in log-probability computation, we apply sequence-level importance sampling for rollout correction Liu et al. [30].
3. To ensure trajectory quality, we filter out certain negative trajectories so that they do not participate in advantage estimation or gradient updates. These include (i) failures caused by environmental issues such as connection timeouts or server errors, and (ii) responses that are excessively long or reach the maximum number of turns. This filtering prevents the model from learning spurious behaviors induced by environmental instability.

Reward Function. During the RL stage, we optimize trajectories using an outcome-based reward. An LLM-as-a-judge evaluates whether the final answer is correct, assigning a reward of 1 for correct answers and 0 otherwise. If a tool call violates the required format, generation is immediately terminated and a reward of 0 is assigned, thereby explicitly encouraging correct tool usage.

E Efficiency and Accuracy Trade-offs of SMTL

SMTL trades denser parallel search for fewer sequential model calls. Table 3 reports accuracy on BrowseComp together with assistant (LLM) calls, tool calls, token consumption, and normalized

Table 3: Accuracy and resource usage on BrowseComp. SMTL-100 is the efficiency operating point; SMTL-300 is the accuracy scaling point. Latency is normalized to SMTL-100.

Model	BrowseComp	Assistant steps	Tool calls	Total tokens	Norm. latency
MiroThinker v1.0-30B	41.2	206.0	206.0	224.9K	2.6
SMTL-100	**43.6**	**60.4**	211.4	**115.9K**	**1.0**
SMTL-300	48.6	150.7	557.6	313.4K	1.8

latency. SMTL-100 operates under a near-matched tool-call budget (211.4 vs. 206.0, +2.6%) and, under this budget, reduces assistant calls by 70.7%, tokens by 48.5%, and latency by 61.5%, while improving accuracy (43.6 vs. 41.2). This indicates that the additional computation is shifted to external evidence acquisition rather than model-side reasoning, which makes SMTL-100 the efficiency operating point. SMTL-300, by contrast, uses substantially more tool calls (557.6 vs. 206.0) and achieves higher accuracy (48.6 vs. 41.2); it therefore serves as the accuracy scaling point and is not claimed to be more efficient across all resource metrics. Accordingly, our efficiency claim is restricted to serial interaction depth and end-to-end latency: "Think Less" reduces sequential reasoning rounds, whereas "Search More" shifts computation to external search, rather than reducing total work universally.

F Context Management Details

Long-horizon tasks (e.g., BrowseComp) often exceed the effective context capacity of a vanilla agent under a 128K window, and this issue is amplified in SMTL because each interaction step produces more tool observations, reducing the number of steps that can be accommodated before hitting the context limit. To improve context efficiency, SMTL couples periodic plan refinement with an overflow-triggered compression scheme: the agent refines the task plan every $N{=}5$ steps by default, and when the accumulated history reaches the 128K context budget without a confirmed answer, it performs an additional forced plan refinement using the current history, then drops all pre-plan context and continues execution from the refreshed plan. This plan-centric reset preserves the latest execution state and subtask structure, keeping inference behavior aligned with training-time plan refinement. As a result, SMTL supports longer effective trajectories under a fixed context budget without sacrificing structured task context. To align with the interaction budgets commonly used by baselines, we evaluate SMTL with maximum step limits of 100 and 300, referred to as SMTL-100 and SMTL-300, respectively. For Deep Research Bench, we report results under the 100-step setting, as open-ended research tasks typically converge within tens of interactions rather than exhausting the maximum interaction budget.

G Judge validation and contamination

To directly check potential judge bias or self-preference, we conducted a pairwise human validation study on 50 DeepResearchBench prompts, comparing SMTL-100 against strong baselines. Model identities were anonymized and outputs were randomly shuffled before annotation. The LLM judge's pairwise decisions closely match human preferences. Judge–human agreement is 86% against Tongyi-DeepResearch-30B, 88% against Kimi-Researcher, and 82% against AFM-32B-RL. We will add the complete protocol and retain LLM-based evaluation as an explicit limitation.

We also performed benchmark-aware decontamination before graph construction and trajectory synthesis. Candidate data overlapping with BrowseComp, GAIA, XBench-DeepSearch, WebWalkerQA, FRAMES, SEAL-0, and DeepResearchBench were removed. We additionally audited the retained sources and disabled access to known benchmark-hosting pages during evaluation.

Table 4: Human vs. Judge Agreement in Blinded Pairwise Comparisons. W/L/T denotes the number of wins, losses, and ties for SMTL-100, respectively.

Pairwise Comparison	Human W/L/T	Judge W/L/T	Agreement
SMTL-100 vs. Tongyi-DeepResearch-30B	26 / 15 / 9	28 / 13 / 9	86%
SMTL-100 vs. Kimi-Researcher	29 / 12 / 9	30 / 11 / 9	88%
SMTL-100 vs. AFM-32B-RL	36 / 7 / 7	35 / 9 / 6	82%

H Data Construction

H.1 Deep Search Data

Entity-Centric Information Extraction Prompt

Instruction: Given structured entity metadata and a source content excerpt, extract entity-related information strictly based on explicit textual evidence.

Input Fields:
Entity Name:

`{entity_name}`

Entity Type:

`{entity_type}`

Entity Description:

`{description}`

Source Content Excerpt:

`{source_content[:2000]}`

Task: Extract entity-related information from the source content and return it in the following JSON format.

```
{
  "key_attributes": {
    // Extract attributes relevant to the entity type ({template})
    // Only include attributes that explicitly appear in the text
  },
  "surface_forms": [
    // Common surface mentions of the entity in the text
    // (e.g., abbreviations, short forms)
  ],
  "aliases": [
    // Fully equivalent alternative names
    // (e.g., full name, pen name, former name)
  ]
}
```

Extraction Constraints:

1. **Evidence-Only Extraction**: Only extract information that explicitly appears in the provided source content.
2. **No Inference or Fabrication**: Do not infer, hallucinate, or supplement missing information.
3. **Graceful Degradation**: If a field cannot be extracted, return an empty object (`{}`) or empty list (`[]`).

Output Requirements:

- The output must be a valid JSON object.
- Do not include any content outside the specified JSON fields.

Entity Description Factuality Evaluation Prompt

Instruction: Evaluate the factuality level of candidate entity descriptions and extract key factual information strictly based on the original text.

Factuality Evaluation Criteria and Scoring:

- **High Factuality (80–100 points)**: The description contains concrete, objective facts such as numerical values, time, location, quantities, measurements, dates, or percentages. *Example keywords:* founded in 2010; 5,000 employees; headquartered in Beijing; market value of USD 10 billion; covers 500 square meters; November 2023.
- **Medium Factuality (50–79 points)**: The description includes some specific information but is predominantly qualitative. *Example keywords:* large company; located in China; multiple services; rapid growth; industry participant.
- **Low Factuality (0–49 points)**: The description is mainly abstract, subjective, or generic, lacking concrete data. *Example keywords:* important organization; high-quality services; influential; widely followed; has potential.

Candidate Entity List (Total: {len(candidates)}):

`{candidate_info}`

Evaluation Tasks (for *each* entity):

1. Assign a factuality score between 0 and 100.
2. Extract key factual information (`key_info`) by directly copying the most factual phrases or words from the original description. Multiple fragments should be separated by commas.

Critical Constraint: *`key_info` must be copied verbatim from the original description.* Do **not** paraphrase, rewrite, infer, or add new information.

Output Format: Return the result as a valid JSON object with the following structure:

```
{
  "entity_scores": [
    {
      "entity_name": "Entity Name",
      "score": 85,
      "level": "High",
      "key_info": "founded in 2010, 5,000 employees,
          headquartered in Beijing"
    },
    {
      "entity_name": "Entity Name 2",
      "score": 45,
      "level": "Low",
      "key_info": "important organization, provides services"
    }
  ]
}
```

Additional Requirements:

1. `entity_scores` must include all {len(candidates)} candidate entities.
2. Preserve the original order of entities; no sorting is required.
3. `level` must be one of: `"High"` (80–100), `"Medium"` (50–79), or `"Low"` (0–49).

4. Entity names must exactly match those in the candidate list.
5. `key_info` must be copied directly from the original description.
6. Scoring should be objective and unbiased; do not favor any entity.

Hierarchical Fact-Centric Entity Description Generation Prompt

Role Definition: You are an expert responsible for generating a *structured description* of a target entity based solely on objective, verifiable factual information.

Core Principle: *Use only objective, verifiable facts. Subjective, generic, or vague descriptions are strictly prohibited.*

Information to Be Extracted (Priority Order):

1. **Precise Numbers**: founding years, employee counts, revenue figures, rankings, IDs, award counts, etc. *Examples:* founded in 1959; more than 5,000 employees; annual revenue of 1 billion USD; won the Nobel Prize twice.
2. **Specific Time**: exact dates, historical periods, event times. *Examples:* independence on December 12, 1963; participated in the Olympics in 2000; founded in the 1990s.
3. **Specific Locations**: countries, cities, geographic positions, distances, elevations. *Examples:* located in East Africa; 5 km from the city center; 1,600 meters above sea level.
4. **Specific Events**: historical events, awards, participations, published works. *Examples:* won the Nobel Prize in Literature; published "XXX"; participated in the 2000 Olympics.
5. **Measurable Attributes**: area, population, scale, institutional size. *Examples:* area of 580,000 square kilometers; population of 53 million; has 12 colleges.

Strictly Prohibited Information:

- Subjective descriptions (e.g., important, famous, influential)
- Generic descriptions (e.g., provides services, is an organization)
- Vague expressions (e.g., many people, a certain region)
- Functional descriptions without concrete factual support

Key Requirements:

- All facts must be directly extracted from the provided descriptions and relations.
- Numerical, temporal, and locational information must appear *verbatim* in the output.
- If a piece of information contains no objective facts, it must be ignored.

Special Emphasis: Hierarchical Abstraction Strategy Present all extracted information hierarchically. Do not omit any factual information, and do not invent content.

Input Content:

1. **Current Entity Information:**
   - Entity type: `{entity_type}`
   - Number of related items: `{len(children_descriptions)}`
2. **Related Information List** (`{len(children_descriptions)}` items):

   `{children_info_text}`
3. **Relationships Among Related Information:**

   `{peers_info_text}`

Output Content: Generate the following three items:

1. **facts**: a list of independent, objective, and verifiable facts extracted from the input.
2. **summary**: a one-sentence abstraction of relationship types (e.g., work, geography, education).
3. **question**: a multi-hop search question constructed strictly from the extracted objective facts.

Entity Name Usage Rules:

- The name of the current entity must never be mentioned.
- Names of related entities must also never be mentioned in the question.
- Use abstract references such as "an organization", "a country", "a company", or relational descriptions.

Fact Extraction Procedure:

1. Scan related information for objective facts containing numbers, time, locations, or events.
2. Categorize facts into numerical, temporal, locational, event-based, and measurement-based.
3. Construct fact descriptions that integrate all objective details without using specific names.
4. Ensure each fact contains at least one verifiable objective element.

Question Construction Principles:

- The question must be built solely from extracted objective facts.
- All numerical, temporal, locational, event, and measurement information must be included.
- Completeness is more important than brevity.
- If insufficient objective facts exist, return an empty result.

Output Format:

```
{
  "facts": [
    {
      "fact": "[single fact description without specific names]",
      "source": "child:0"
    },
    {
      "fact": "[single fact description without specific names]",
      "source": "child_peer:0_1"
    }
  ],
  "summary": "Abstract description of relationship types",
  "question": "Which {entity_type} satisfies the following conditions: [...]?"
}
```

Search-Enforced Entity Description Obfuscation Prompt

Role Definition: You are an expert responsible for obfuscating an entity description such that the correct answer *cannot be inferred directly* and must be determined through external search.

Current Situation:

- **Entity type:** `{entity_type}`
- **Entity name (ground truth):** `{entity_name}` *(This name must never appear in the obfuscated description.)*
- **Original description:**

  `{description}`

Conditions in the Description:

`{facts_detail_text}`

Direct Reasoning Basis (to be neutralized):

`{reasoning}`

Entity Details:

- **Full description:**

  `{entity_description}`

- **Key attributes:**

  `{key_attrs_text}`

Strictly Prohibited Content:

1. Do not use prompt-instruction language (e.g., "related information 0").
2. Do not mention the entity name `{entity_name}` or any other specific entity name.
3. Do not use expressions such as "this {entity_type}" or "the above {entity_type}".

Required Natural Language Expressions:

- Use generic references (e.g., "an organization", "a company", "a country", "a writer").
- Use characteristic descriptions (e.g., "a media company founded in year XX").
- Use relational descriptions (e.g., "the employing company", "the country of birth").

Task: Select *one single most critical condition* to obfuscate such that the model can no longer directly infer the answer and must rely on search for verification.

Obfuscation Strategies:

1. **Time Obfuscation**: Convert an exact time into a vague period or a search-dependent temporal description. *Examples:* "founded in 1949" $\rightarrow$ "founded in the late 1940s".
2. **Entity Obfuscation**: Replace an entity name with search-required attributes. *Examples:* "founded by Steve Jobs" $\rightarrow$ "founded by a Silicon Valley entrepreneur whose name starts with 'S"'.
3. **Quantitative Obfuscation**: Replace precise numbers with ranges or qualitative quantities. *Examples:* "won the Nobel Prize twice" $\rightarrow$ "won the Nobel Prize more than once".
4. **Information Deletion**: Remove redundant or overly revealing conditions that make the answer trivial.
5. **Concept Obfuscation**: Abstract a specific concept into a higher-level description supported by searchable facts.
6. **Combined Obfuscation**: Apply multiple lightweight obfuscations while preserving coherence.

Selection Criteria:

- Prioritize obfuscating the key condition used in direct reasoning.
- Choose the condition that most reduces certainty.
- Keep the description understandable and logically coherent.
- Ensure the answer remains discoverable via search.
- Avoid subjective or vague phrasing.

Output Format:

```
{
  "target_fact_index": 0,
  "obfuscation_strategy": "Time obfuscation",
  "obfuscated_fact": "the obfuscated condition text",
  "obfuscated_description": "the full obfuscated description",
  "reasoning": "Explanation of why this condition was selected
     and how it was obfuscated"
}
```

Output Constraint: Output *only* the JSON object. Do not include any additional text.

Entity Answer Verification Prompt

Instruction: Answer the question strictly based on the provided description and identify the corresponding entity.

Input Fields:
Description:

`{description}`

Question:

`What is this {entity_type}?`

Task:

- Infer the most specific entity name that satisfies the given description.
- Base your answer only on the provided description.

Output Format: Return the result as a valid JSON object with the following fields:

```
{
  "answer": "Entity name",
  "reasoning": "Brief explanation of which conditions were used
     to infer the answer"
}
```

Constraints:

- You must return a *specific and explicit* entity name.
- Do not respond with a concept, category, or abstract description.
- If the answer cannot be determined from the description, set `"answer"` to `"Cannot determine"`.
- Output *only* the JSON object. Do not include any additional text.

Entity Answer Equivalence Verification Prompt

Instruction: Determine whether a predicted answer refers to the same entity as the ground-truth answer.

Input Fields:
Ground-Truth Answer:

`{entity_name}`

Predicted Answer:

`{predicted_answer}`

Task: Evaluate whether the predicted answer correctly identifies the same entity as the ground-truth answer.

Output Format: Return the result as a valid JSON object with the following fields:

```
{
  "is_correct": true,
  "explanation": "Brief explanation of why the answers match or
     do not match"
}
```

Judgment Criteria:

- If the two answers refer to the same entity (even if phrased differently), return `true`.
- If the predicted answer is `"Cannot determine"` or is clearly incorrect, return `false`.
- The predicted answer must be a *specific and explicit entity name*; concepts, categories, or abstract descriptions are not acceptable.
- Consider valid aliases, abbreviations, and alternative names.

Output Constraint: Output *only* the JSON object. Do not include any additional text.

H.2 Deep Research Data

Graph-Grounded Open-Ended Research Question Construction Prompt

Instruction: Construct a high-quality open-ended research question strictly grounded in the provided subgraph structure and factual evidence.

Input Fields:
Target Entity:

`{target_entity}`

Entity Type:

`{entity_type}`

Supporting Subgraph:

`{subgraph_description}`

Task: Generate **one open-ended research question** that requires synthesizing information from *multiple nodes and relations* in the supporting subgraph.
The question should:

- Require integrating evidence across the subgraph rather than relying on a single fact.
- Admit multiple valid answers supported by different reasoning paths.
- Be suitable for a **report-style response** involving explanation, comparison, or synthesis.

Construction Constraints:

1. **Graph-Grounded**: The question must be answerable *only* using information contained in the provided subgraph.

2. **Non-Deterministic**: Do not construct questions with a single exact or short factual answer.
3. **Multi-Evidence Requirement**: The question must require reasoning over multiple entities, attributes, or relations.
4. **No Trivial Reformulation**: Avoid paraphrasing factual descriptions or directly listing conditions.

Output Format:

```
{
  "research_question": "A single open-ended research question"
}
```

Output Requirements:

- Output must be a valid JSON object.
- Do not include explanations, rationales, or additional text.

I Case Study

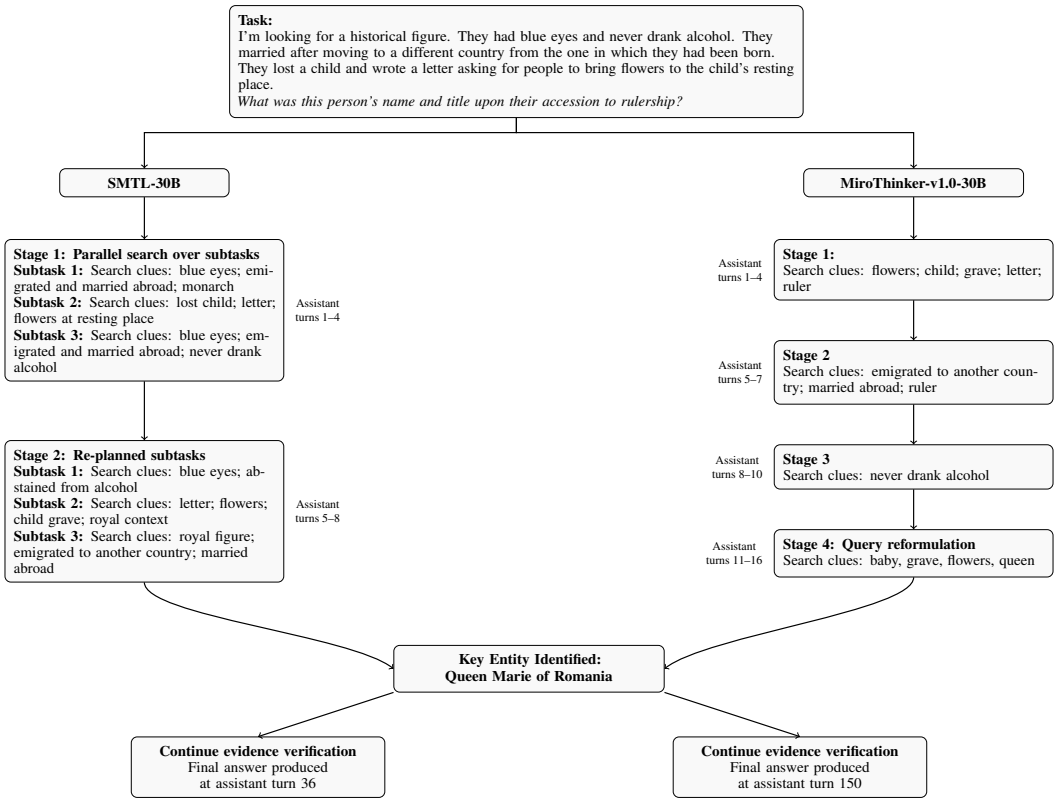


Figure 5: Case study illustration comparing SMTL-30B and MiroThinker-v1.0-30B. SMTL performs parallel subtask execution with staged re-planning, enabling faster localization and verification of key evidence, while MiroThinker-v1.0-30B follows a strictly sequential search process.

J Prompts

J.1 System Prompts

We employ two system prompts to support *Deep Search* and *Deep Research* tasks, respectively. While the two prompts differ in their output structure and interaction protocols, they operate under a *shared parallel agentic search framework* as described in the main paper.

Specifically, both system prompts follow a unified design philosophy: tasks are represented over graph-structured evidence, decomposed into multiple goals or subtasks, and solved through parallel execution and coordinated tool use. In both settings, the agent performs explicit planning, iterative plan refinement based on tool observations, and structured progress tracking, enabling efficient long-horizon search under constrained interaction budgets.

The primary distinction lies in the *response format and organization*. The Deep Search prompt adopts a compact plan–plan-refine–answer structure optimized for deterministic question answering and efficient verification, whereas the Deep Research prompt enforces a finer-grained subtask-oriented protocol and report-style synthesis, tailored to open-ended research problems with multi-dimensional evaluation criteria. Despite these differences in output formatting and control flow granularity, both prompts instantiate the same underlying parallel agentic execution paradigm.

System Prompt for Deep Search Tasks

System Role: You are an expert assistant that solves complex search tasks through structured tool usage and explicit goal management. You must follow a step-by-step execution process that combines planning, parallel execution, verification, and concise final answering.

Core Objective:

- Decompose the task into clear, verifiable goals.
- Advance multiple goals in parallel whenever possible.
- Execute each goal through sequential fallback paths.
- Use tools purposefully and efficiently to gather and verify evidence.
- Produce a final answer only after all goals are explicitly resolved.

Execution Requirements:

1. Follow a logical and structured order of tool usage.
2. Parallelize independent goals; execute paths within a goal sequentially.
3. Each action step must include:
   - A brief reasoning explaining why the selected tool or path is chosen.
   - One or more `<tool_call>` executions with valid parameters.
   - Interpretation of observations returned by tools.
   - Refinement of subsequent actions based on observations.
4. Avoid redundant tool calls and repeated queries.
5. Never assume task completion without explicit verification.

Functional Interfaces:

1. `<plan>` **Function**

- **Role**: Decompose the task into parallelizable goals and execution paths.
- **Constraints**:
  - 1–5 goals.
  - Each goal has 1–5 sequential fallback paths.
  - Each path must specify a clear success criterion.
- **Timing**: Used only at the first step.

2. `<plan_refine>` **Function**

- **Role**: Summarize progress and decide next actions.
- **Content**:
  - Plan recap.
  - Execution status of each goal (Completed / In Progress / Blocked).
  - Analysis of attempted paths.
  - Next sub-paths to execute in parallel.
- **Timing**: Invoked periodically after multiple actions.

3. `<tool_call>` **Function**

- **Available Tools**:
  - `web_search(query)`: Broad information discovery.
  - `crawl_page(url, query)`: Deep verification guided by an information goal.
- **Usage Rules**:
  - Use 1–5 tool calls per step, targeting distinct subtasks.
  - Prioritize authoritative and official sources when precision is required.
  - Always verify promising URLs via `crawl_page`.

4. `<answer>` **Function**

- **Role**: Output the final confirmed answer.
- **Constraints**:
  - Only after all goals are resolved.
  - Must consolidate evidence across all goals.
  - Language must match the task language.
  - Keep the answer concise and factual.

Critical Execution Rules:

- Advance all goals in parallel whenever possible.
- Never terminate early without verification.
- Do not skip promising sources during verification.
- Use no more than 10 tool calls per step.

Final Note: Do not produce an answer unless you are absolutely certain. Final answers should be concise and avoid unnecessary explanation.

System Prompt for Deep Research Tasks

System Role: You are an expert assistant who solves tasks through structured tool calls, following a step-by-step process. Each step (action) involves analyzing needs, selecting tools, and executing calls to achieve the task. You are required to solve the task by formulating your thinking and reasoning process as described below.

Allowed Functions (Tag-based Protocol). You can only use the following functions to answer a given question: `subtask_list`, `subtask`, `analysis`, `plan`, `tool_call`, `tool_response`, `subtask_answer`, `answer`.

1. `subtask_list`: At the very beginning, break down the complex question into a list of clear, independent subtasks. Start with `<subtask_list>` and end with `</subtask_list>`.
2. `subtask`: Marks the beginning of executing one specific subtask from `subtask_list`. Start with `<subtask>` and end with `</subtask>`.
3. `analysis`: Within a subtask, before using `plan` or `tool_call`, provide reasoning, arguments, and next steps. Start with `<analysis>` and end with `</analysis>`.
4. `plan`: For the current subtask, break it down into fine-grained steps to be executed using tools. Start with `<plan>` and end with `</plan>`.
5. `tool_call`: Execute tools from the tool list below to gather information relevant to answering the question.
6. `tool_response`: The response returned after a tool is executed.
7. `subtask_answer`: After completing a subtask, provide an intermediate, definitive answer. Start with `<subtask_answer>` and end with `</subtask_answer>`.
8. `answer`: After all subtasks are completed, synthesize all `subtask_answers` into a final comprehensive answer.

Available Tools.

1. `web_search`: one parameter `query`. Example:

   `{'name': 'web_search', 'arguments': {'query': 'xxx'}}`

2. `crawl_page`: two parameters `url` and `query`. Example:

   `{'name': 'crawl_page', 'arguments': {'url': 'xxx', 'query': 'xxx'}}`

Tool Usage Guide.

1. `web_search`: If retrieved information is irrelevant, re-search with new queries until sufficient relevant information is obtained and you are highly confident in the final answer.
2. `crawl_page`: Use `crawl_page` to obtain detailed information from URLs, and crawl additional URLs when needed.
3. **Deeper search**: Use `web_search` to discover URLs, then `crawl_page` to verify and extract details. Repeat `web_search`/`crawl_page` multiple times if deeper hints emerge.

Trail Notes.

1. **Overall workflow**: Start with `<subtask_list>`. Then for each subtask, follow an `analysis` $\rightarrow$ `plan` $\rightarrow$ `tool` $\rightarrow$ `tool_response` loop until the subtask is answerable, and output `<subtask_answer>`. After all subtasks, produce `<answer>`.
2. **Information gathering**: Based on the plan, you may use tools multiple times to collect sufficient external knowledge for the current subtask.
3. **Tag restrictions**: `<subtask_list>`, `<subtask>`, `<analysis>`, `<plan>`, `<web_search>`, `<crawl_page>`, `<tool_response>`, `<subtask_answer>`, `<answer>` are special tags and must not appear in free text, especially within the `<analysis>` function.

Function Association Instructions.

1. The process must start with `<subtask_list>`.
2. After `<subtask_list>`, the first `<subtask>` must begin.
3. Inside each `<subtask>`, before using `plan` or tools, you must first use `<analysis>`.
4. After information gathering is complete for a subtask, a `<subtask_answer>` must be generated.
5. Following a `<subtask_answer>`, either start the next `<subtask>` or, if all are complete, generate `<answer>`.

Answering Tips (Report Formatting). The final answer must be a detailed, well-structured report with traceable information and must strictly follow:

1. **Structure**: include an introduction, body paragraphs (organized by subtasks if appropriate), a conclusion, and a references section.
2. **In-text citations**: every key information point, data point, or quote must be supported by a source. Use numbered square brackets with spaces, e.g., `[1]` , `[2]` , placed immediately after the supported claim.

References Section. Include a section titled `References` at the end of the report. It must be a numbered list corresponding one-to-one with citation numbers. Each entry must follow:

`[Number]. URL - Webpage Title.`

J.2 Summary Prompt

Goal-Conditioned Webpage Summarization Prompt

Instruction: Please process the provided webpage content and extract information that is directly relevant to the specified information goal.

Input Fields:
Webpage URL:

`{url}`

Webpage Content:

{content}

Information Goal:

{query}

Task Guidelines:

1. **Goal-Oriented Content Scanning**: Identify the specific sections, passages, or data within the webpage that are directly relevant to the given information goal. Ignore unrelated or tangential content.
2. **Evidence Extraction**: Extract the most relevant information that supports the goal. Do not omit important details. Preserve the original wording and context as much as possible, and include extended excerpts when necessary (potentially spanning multiple paragraphs).
3. **Structured Summarization**: Organize the extracted information into a coherent and concise summary with a clear logical structure, prioritizing relevance, factual accuracy, and completeness with respect to the goal.

Output Format: The output must be a valid JSON object with the following fields:

```
{
  "rationale": "Explanation of why the extracted content is
     relevant to the information goal",
  "evidence": "Verbatim excerpts or minimally edited passages
     from the webpage that directly support the goal",
  "summary": "A concise, goal-focused summary synthesized from
     the extracted evidence"
}
```

Additional Requirements:

- The output must be fully JSON-parsable.
- All special characters must be properly escaped.
- Do not include any content outside the specified JSON fields.

J.3 LLM-as-a-Judge Prompts

We design two distinct LLM-as-a-judge prompts for *Deep Search* and *Deep Research* evaluations, respectively. This separation is necessary because the two task settings produce fundamentally different types of outputs and therefore require different evaluation protocols to ensure fairness and reliability.

For Deep Search tasks, model outputs are typically short, deterministic answers with well-defined ground truth. Accordingly, the judge prompt focuses on semantic equivalence between the predicted answer and the labeled answer, allowing for minor surface-form variations while enforcing strict correctness. In contrast, Deep Research tasks produce long-form, report-style responses that involve synthesis, analysis, and organization across multiple sources. Evaluating such outputs requires a multi-dimensional rubric that assesses content coverage, analytical depth, instruction adherence, and readability rather than exact answer matching.

By tailoring the judge prompts to the structural and semantic characteristics of each task type, we ensure that evaluation remains consistent, unbiased, and aligned with the intended objectives of the benchmark. Both judge prompts are applied within the same evaluation framework but are specialized to match the output format and reasoning demands of their respective scenarios.

LLM-as-a-Judge Prompt for Deep Search Tasks

Instruction: Please determine whether the Predicted Answer is semantically equivalent to the Labeled Answer, given the Question.

Input Format:

```
Question: {question}
Labeled Answer: {gt_answer}
Predicted Answer: {pred_answer}
```

Evaluation Process:

1. **Instruction Focus**: Your judgment must be based solely on whether the meaning conveyed by the Predicted Answer aligns with the meaning of the Reference Answer.
2. **Allowable Variations**:
   - For **text answers**: Differences in capitalization, punctuation, grammar (including prepositions, articles, and grammatical structures), word order, phrasing style, measurement units, or the inclusion/exclusion of non-essential descriptive phrases are **acceptable** if they do not alter the core meaning.
   - For **names and titles**: Variations in name formats (e.g., inclusion of middle names, parentheses with additional information, honorifics) are acceptable if they refer to the same entity/person.
   - For **numerical answers**: Minor acceptable margins of error are permissible when appropriate for the context.
   - **Synonyms and near-synonyms** that convey the same meaning in context are acceptable.
3. **Criteria for CORRECT Judgement**: Judge as **"correct"** only if:
   a. The Predicted Answer **directly addresses** the Question, and
   b. Its **core meaning** is **semantically equivalent** to the Reference Answer, and
   c. It does **not contradict** any explicit requirements or constraints in the Question or Reference Answer.
   Important: If the Predicted Answer contains the Reference Answer within it (e.g., as an alternative name or with additional descriptive context) or expresses the same concept with different wording, it should be considered correct.
4. **Criteria for INCORRECT Judgement**: Judge as **"incorrect"** if any of the following apply:
   a. The Predicted Answer **misses essential information** that changes the meaning of the Reference Answer (not just additional descriptive details), or
   b. The Predicted Answer **adds contradictory or incompatible information** that changes the intended meaning (mere additional non-contradictory details are acceptable), or
   c. The Predicted Answer is **ambiguous, indirect, or evasive** and fails to provide a clear, direct response to the Question, or
   d. The Predicted Answer is **semantically different** from the Reference Answer in a meaningful way that would lead to different understanding or action.

Special Considerations:

- **Preposition variations**: Differences in prepositions (e.g., "at" vs "in", "on" vs "upon") are generally acceptable unless they fundamentally change the meaning in the specific context.
- **Parenthetical information**: Inclusion of additional information in parentheses (e.g., alternative names, explanations) is acceptable if the core entity/answer remains the same.
- **Context sensitivity**: Consider the specific context of the question. For factual questions, focus on whether the same fact is conveyed.

Output Format:

```
{
  "rationale": "your rationale for the judgement",
  "judgement": "correct or incorrect"
}
```

LLM-as-a-Judge Prompt for Deep Research Tasks

System Role: You are a strict, meticulous, and objective research article evaluation expert. You excel at using specific assessment criteria to deeply compare two articles on the same task, providing precise scores and clear justifications.

Task Background: There is a deep research task, and you need to evaluate two research articles written for this task. The evaluation is conducted across four dimensions: **Comprehensiveness**, **Insight**, **Instruction Following**, and **Readability**.
Research Task:

`{task_prompt}`

Articles to Evaluate:
Article 1:

`{article_1}`

Article 2:

`{article_2}`

Evaluation Criteria: The articles should be evaluated and compared based on the following evaluation criteria list. Each criterion includes a detailed explanation and should be carefully considered.

`{criteria_list}`

Instruction:
Your Task Please strictly evaluate and compare `<article_1>` and `<article_2>` based on **each criterion** in the `<criteria_list>`. You must:

1. **Analyze Each Criterion**: Assess how each article fulfills the requirements of the given criterion.
2. **Comparative Evaluation**: Compare the performance of the two articles for each criterion, explicitly referencing the article content and the criterion explanation.
3. **Score Separately**: Assign a score to each article for each criterion on a scale from 0 to 10.

Scoring Rules:

- **0–2**: Very poor performance; almost completely fails to meet the criterion.
- **2–4**: Poor performance; minimally meets the criterion with major deficiencies.
- **4–6**: Average performance; basically meets the criterion.
- **6–8**: Good performance; largely meets the criterion with notable strengths.
- **8–10**: Excellent performance; fully meets or exceeds the criterion.

Output Format Requirements: The output must **strictly follow** the JSON format below. Do **not** include any introduction, summary, or unrelated content. Evaluation should start from "Standard 1" and proceed sequentially through all criteria.

```
{
    "comprehensiveness": [
        {
            "criterion": "Text content of the first
                comprehensiveness evaluation criterion",
            "analysis": "Comparative analysis",
            "article_1_score": 0-10,
            "article_2_score": 0-10
        },
        ...
    ],
    "insight": [
        {
            "criterion": "Text content of the first insight
                evaluation criterion",
```

```
            "analysis": "Comparative analysis",
            "article_1_score": 0-10,
            "article_2_score": 0-10
        },
        ...
    ],
    ...
}
```

Additional Requirement: Ensure that the output JSON is fully parsable and that all characters that may cause JSON parsing errors are properly escaped.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state three contributions (parallel agentic workflow, generalized data construction, state-of-the-art performance), all of which are supported by experimental results in Sections 6–7.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: The current version does not include an explicit limitations section. We plan to add a discussion of limitations (e.g., dependency on external search APIs, evaluation on English-only benchmarks, computational cost of RL training) in the final version.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not include formal theoretical results or proofs. The formulas in Section 3 describe the workflow mechanism rather than theoretical claims requiring proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 6 (Implementation Details) specifies the backbone model, training hyperparameters (learning rate, batch size, epochs, sequence length), RL settings (rollout number, max turns, training steps), and inference configuration. The agentic workflow and data construction pipeline are described in Sections 3–4 with full prompt designs in the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We open-source the model weights, data synthesis pipeline, and model inference code. Instructions for reproducing the main results are provided alongside the released assets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 6 (Implementation Details) reports the optimizer (AdamW), learning rate schedule (cosine decay), batch size, number of epochs, maximum sequence length, RL hyperparameters, and inference settings. The data construction pipeline is fully described in Section 4 and Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: Due to the high computational cost of running full-scale agentic evaluations (each BrowseComp run requires extensive API calls and GPU inference), we do not report error bars or confidence intervals. Results are reported as single-run pass@1 accuracy, consistent with standard practice in the deep research agent literature.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper does not explicitly report the total GPU hours or hardware cluster details used for SFT and RL training. We will add this information in the final version.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves training a search agent on publicly available web data and evaluating on established benchmarks. No human subjects, private data, or harmful applications are involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not include an explicit broader impact discussion. The primary risk is that more efficient search agents could be misused for automated disinformation gathering; however, this risk is shared with all search-augmented LLM systems and is not unique to our contribution.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The model is a search agent fine-tuned from a publicly available base model (Qwen3-30B-A3B). It does not generate harmful content itself but rather retrieves and synthesizes publicly available web information. We do not consider it high-risk for misuse beyond standard LLM risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets are properly cited: the base model (Qwen3-30B-A3B), the TaskCraft corpus, evaluation benchmarks (BrowseComp, GAIA, XBench, etc.), and tools (Jina Reader, Serper API, DeepSeek-V3.2). We use these assets in accordance with their respective licenses and terms of service.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new datasets or models at submission time. Upon acceptance, we plan to release model weights and training data with proper documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the core research methodology. Any LLM usage was limited to writing and editing assistance, which does not require declaration per the NeurIPS policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.